

RESEARCH

Open Access

# Sleep/wake scheduling scheme for minimizing end-to-end delay in multi-hop wireless sensor networks

Babar Nazir<sup>1\*</sup>, Halabi Hasbullah<sup>1</sup> and Sajjad A Madani<sup>2</sup>

## Abstract

We present a sleep/wake schedule protocol for minimizing end-to-end delay for event driven multi-hop wireless sensor networks. In contrast to generic sleep/wake scheduling schemes, our proposed algorithm performs scheduling that is dependent on traffic loads. Nodes adapt their sleep/wake schedule based on traffic loads in response to three important factors, (a) the distance of the node from the sink node, (b) the importance of the node's location from connectivity's perspective, and (c) if the node is in the proximity where an event occurs. Using these heuristics, the proposed scheme reduces end-to-end delay and maximizes the throughput by minimizing the congestion at nodes having heavy traffic load. Simulations are carried out to evaluate the performance of the proposed protocol, by comparing its performance with S-MAC and Anycast protocols. Simulation results demonstrate that the proposed protocol has significantly reduced the end-to-end delay, as well as has improved the other QoS parameters, like average energy per packet, average delay, packet loss ratio, throughput, and coverage lifetime.

**Keywords:** wireless sensor network, sleep/wake scheduling, MAC, duty cycle, energy efficiency; routing

## 1. Introduction

The latest advances in distributed computing and micro electro mechanical systems have enabled in the past few years the emergence of a variety of wireless sensor network applications comprising military [1], disaster management [2], building, health [3], environment, industry, and domains. Wireless sensor network is a network of spatially distributed sensor nodes equipped with sensing, computing, power, and communication modules to monitor a certain phenomenon such as environmental data or object tracking [4,5]. The nodes in such networks are characterized by limited power, processing, and memory resources. As the sensor nodes are powered by batteries, it is difficult to replace or recharge these batteries because of cost (e.g., cost of batteries and labor) or geographic (e.g., difficult or unfriendly terrain) reasons.

A sensor node consumes battery power in the following four operations: sensing data, receiving data, sending

data, and processing data. Generally, the most energy consuming component is the RF module that provides wireless communications. Consequently, out of all the sensor node operations, sending/receiving data consumes more energy than any other operations. The energy consumption for transmitting 1 bit of data on the wireless channel is equivalent to the energy required to execute thousands of cycles of CPU instructions [5]. Therefore, efficient use of energy in WSN communication protocols extends the network lifetime. Hence, any MAC, network, and transport layer protocols designed for WSN should give due consideration to the efficient use of RF module by minimizing MAC collision, control message overhead in routing, efficient sleep/wake scheduling and so on. In addition, during protocol design, the limited resources of sensor nodes should also be considered, which includes low processing power, less memory, short-range communication, and low sensing power.

Generally, WSNs operate for a long time in idle mode and only occasionally send data. The energy consumption of listening to the idle channel is equivalent to its

\* Correspondence: babarnazir@gmail.com

<sup>1</sup>Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia  
Full list of author information is available at the end of the article

energy consumption when sending or receiving, and much larger than the energy consumption of the sleep mode [4]. To receive data, the receiver must be in high power state, for example, active/listen state; as in sleep state, the radio is in low power mode with the receiving circuitry switched off [6,7]. If the receiver operates at 100% duty cycle, that is, its transceiver is always on; then it would be able to receive the data at the cost of high energy consumption. To reduce the power consumption low duty cycle operations are required. This fact is exploited by sleep-wake scheduling techniques and effort is made to reduce this energy wastage in idle mode by designing low duty cycle operations.

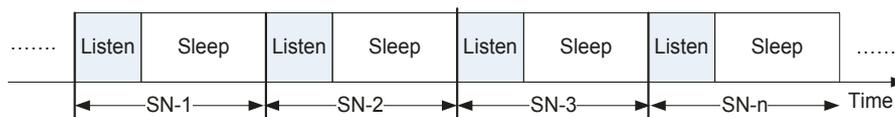
Variety of sleep/wake scheduling protocols has been proposed in the literature. Most of them use a period sleep/wake interval (see Figure 1) and provide effective energy conservation at the cost of delay and throughput. For example, for a source node to transmit data, it has to know the sleep/wakeup schedule of the neighbor node and has to wait for the neighbor to come into the active state. The same is repeated until the data reaches the final destination thus resulting in unprecedented delays. This increase in delay is equal to the product of the number of intermediate forwarders times the length of the wakeup interval [8]. Such increase in end-to-end delay incurred due to latency-energy trade off has the potential to become major problem in many emerging delay-sensitive WSN applications, which require fast response and real-time control.

In most of the WSN sleep/wake protocols, energy awareness is considered as a key design issue to maximize the network lifetime at the cost of latency, delay, and throughput; thus making such design inapplicable in delay sensitive applications. In this article, the delay-minimization problem in sleep/wake scheduling is investigated for event-driven sensor networks in delay-sensitive applications. A distributed and low-complexity solution is presented. The scheme is based on developing schedules based on traffic load requirements of nodes to reduce latency and enhance energy efficiency at the same time. To reduce delay, the proposed protocol does not use a generic sleep/wake schedule for all the nodes, rather it uses a heuristic which maximizes the active duration of the nodes according to their expected traffic load at three different levels. Firstly, the nodes' wake interval is increased as the distance of the node from the BS

decreases as such nodes send their own data as well as act as relay nodes for those away from the BS. Secondly, the wake interval of nodes may also be increased due to their topological importance. Topological importance can be determined by the node's role in the network connectivity. For instance, a node which happens to be cut vertex (e.g., the only node to connect two parts of the network), expects to have greater traffic load as compared with the normal nodes, because the traffic of two sub-network will be forwarded through this node. Such nodes need to have longer wake intervals. Thirdly and lastly, in case of occurrence of an event at a node; the wake interval of that particular node as well as that of its neighbors is also increased in anticipation that more events may occur at the same node or the nearby vicinity.

The main contributions of this paper can be summarized as follows:

- Firstly, the importance of delay minimization problem is discussed and use of variable sleep/wake schedule for different nodes according to their traffic load requirement is advocated, in contrast to general sleep/wake approach.
- Secondly, Sleep/Wake Scheduling Scheme for Minimizing End-to-end Delay (SMED) in Multi-hop Wireless Sensor Network is introduced to address delay minimization problem. To reduce delay, SMED identifies nodes for different sleep/wake according to their traffic load at three different levels based on (a) node's distance from BS, (b) topological importance of node, and (c) handling traffic bursts in the proximity of event occurrence.
- Thirdly and lastly, extensive simulations are carried out to evaluate the performance of the proposed protocol, by comparing its performance with Anycast and S-MAC protocols. The results demonstrated that SMED has successfully minimized the end-to-end delay, as well as has improved the other QoS routing parameters like average energy per packet, packet loss ratio, throughput, and networks lifetime.
- The rest of the article is organized as follows. In Section 2, related work is summarized. In Section 3, Sleep/Wake Scheduling Scheme for Minimizing End-to-end Delay (SMED) in Multi-hop Wireless Sensor Network is proposed. Simulation results are



**Figure 1** Generic periodic sleep and listen schedule for network nodes.

presented in Section 4, and conclusions and future work are offered in Section 5.

## 2. Related work

Over the years, energy efficiency is regarded as one of the critical design issues in WSN. To achieve energy efficiency, many Power Conservation Mechanisms (PCMs) have been proposed [9,10]. Research by Nikolaos [11] gives a comprehensive survey of different energy conservation protocols. It divided PCMs into two main types: active and passive. Active schemes enhance energy efficiency at protocol level (MAC, network, and transport layers). Passive schemes on the other hand rely on energy aware hardware, for example, low power radio and processor [12-15]. When nodes are idle, they go into low power sleep mode (turn off their transceiver) and wakeup (turn on their transceiver) using self timer. Depending upon the decision when to turn on or off the transceiver, the PCM schemes are divided into two categories [11]: fine grain and coarse grain schemes. In fine grain schemes, node turn off their transceiver for one transmission frame, when there is no frame destined to that node. Power-aware multi-access with signaling (PAMAS) [16] is an example of fine grained scheme. In PAMAS, when a node is receiving data from any other node; it sends a busy tone on its control channel to tell the neighbor node when to turn off. A sensor node turns off its transceiver when it does not have data to send and its neighbor starts communication with other nodes. Contrary to this, in the coarse-grain power conservation mechanism, the decision when to turn on or turn off the transceiver is done by a dedicated service above the MAC layer [9]. Coarse-grain mechanism is further divided into two categories: distributed and backbone-based. In the distributed mechanism, the decision when to turn on or turn off the transceiver is taken by the node itself considering its local information and sleeping schedule information of its neighbor. S-MAC [17] is an example of distributed coarse-grain passive protocol and is one of the first MAC layers designed to reduce power consumption in WSNs. In S-MAC, nodes are randomly turned on and turned off to save energy consumption. Traffic destined for these randomly sleeping nodes is temporarily stored by the neighbors, who are active. Sleeping nodes after periodic intervals wake up and receive/save their data from their neighbors. S-MAC enhances energy efficiency to a great extent at the cost of increased delay. In the backbone-based approach, the power conservation module lies at the backbone node. These backbone nodes act as coordinators, do the synchronization of sleeping schedules and also act as proxy for the sleeping sensor nodes. SPAN [18] is an example of a backbone-based coarse-grain conservation scheme, where each node makes local decision to go into sleep or join backbone, based on its own

energy and estimation that how many of its neighbor will benefit from it being awake. In Ref. [19], a scheme based on cross layer design virtual back bone is formed using clustering.

Coarse-grain power conservation mechanisms can further be divided into three main types [11,20]: on-demand, scheduled rendezvous, and asynchronous. In on-demand protocols, a node wakes up only when some other node wants to talk to it. Low power wakeup radio in addition to the main radio is used to address this issue but is restricted by geographic scalability (i.e., the range of the wake-up radio is very limited). In Sparse Topology and Energy Management (STEM) [21], each node sends beacon on the wakeup channel to inform the receiver about its intention to communicate. The receiver node replies by sending an ACK and then the sender sends the data on the data channel. STEM-C [22] sends a wakeup tone before communicating. Unlike a wakeup beacon, which is only received by one receiver, a wakeup tone is received by all the nodes in the neighborhood. In Ref. [23], a Pipelined Tone Wakeup (PTW) scheme is proposed. PTW also uses two different channels, one for wakeup signal and other for data packet propagation. Here, the receiver node periodically wakes up and the sender sends a wakeup tone when it detects any event. With this approach, PTW significantly reduces message latency as compared with STEM. The use of an extra radio (wake-up) has two basic limitations: cost and range.

In the schedule rendezvous approach,<sup>a</sup> the nodes are scheduled to wake-up at the same time when their neighbor nodes wakeup. In this way, a node can have communication with its neighbors as nodes in the same locality have the same schedule/wake interval. The issue with this scheme is that nodes may have to maintain multiple wake-up schedules.

In the asynchronous approach, a node can wake-up at any instance when it wants to communicate. Overlapping between wake intervals of the communicating nodes is ensured. In [24], the authors first use asynchronous approach for IEEE 802.11 *ad hoc* networks. Later, Ref. [25] presents asynchronous wakeup mechanisms for *ad hoc* networks which also can be applied to wireless sensor networks.

Several MAC schemes that focuses on enhancing energy efficiency can also be found abundantly in the literature. Such schemes determine which nodes should be allowed to sleep and vice versa. Some of important efforts are as under.

Reference [26] proposes a linear distance-based sleep scheduling scheme for cluster-based sensor networks. In this scheme, a sensor node decides to go into sleep state based on a probability that is proportional to its distance from the cluster head. But this scheme results in unequal energy consumption of sensor nodes in the cluster.

Reference [27] uses an approach where energy utilization among nodes is balanced by considering the total energy spent in communication and sensing.

In Ref. [28], the Basic Energy Conserving Algorithm (BECA) and the Adaptive Fidelity Energy-Conserving Algorithm (AFECA) are proposed. In the BECA scheme, nodes switch among three states: active, idle, and sleep based on the information obtained from routing or application layer. For instance, a node changes its state from active to idle state if it has nothing to send. AFECA scheme works with on-demand routing protocol. Span is proposed in Ref. [18] to increase the sleep state of the node while keeping the same traffic latency. To do this, some of the nodes never sleep, termed as coordinators. The coordinator plays a pivotal role here and defines the sleep/wake time interval for all the nodes. Furthermore, as it remains active all the time, thus, only the coordinator participates in routing. As the coordinator can deplete all its energy by remaining awake all the time, it becomes the bottleneck.

In Ref. [29], a back-off based sleep scheduling protocol is proposed to decrease the energy consumed by the network while maintaining the same sensing coverage. Here, each node finds out its redundant sensing coverage and decides to go into the sleep state, if sensing coverage can be maintained when it is turned off. A back-off mechanism is used to avoid two or more nodes making sleeping decision at the same time. Each node in the network autonomously and periodically makes decisions on whether to turn on/off itself using local neighbor information. To preserve sensing coverage, a node decides to turn it off when it discovers that its neighbors (sponsors) can help it to monitor its whole working area. A random back-off-based scheme is introduced to avoid blind points, which may appear when two neighboring nodes expect each other's sponsoring.

Dynamic sensor MAC (DSMAC) [30] introduces a dynamic sleeping cycle by extending SMAC, which is based upon network latency and power availability on a node-by-node basis. In PMAC [31], the sleep schedule for the whole network is dynamically made based on throughput and longer sleep periods are used when network utilization is low. In UMAC [32] variable sleep schedules are given to different nodes based on network utilization. Nodes learn the sleep schedule of each of their neighbor nodes and wakeup only to transmit when they know their destination node is awake. In [33], Anycast packet-forwarding scheme is proposed, where each node has multiple next-hop relaying nodes in a candidate set referred as forwarding set. Thus, when a node has data to send, it needs to wait for one specified next-hop neighbor to wake, rather, it forwards the packet to the first node that wakes up in the forwarding set. It reduces the expected one-hop delay.

Through the extensive literature review, it is concluded that in most sleep/wakeup schemes, all nodes have the same generic sleep/wake schedule and each node makes a wake-up decision in isolation, without considering its neighbors in order to save energy. However, as WSNs use multi-hop communication, every node has one designated next-hop relaying node in the neighborhood, thus, to do the transmission, sender node has to wait for the arrival of wakeup time of the next-hop forwarding node. Similarly, next-hop forwarding nodes have to wait for the wakeup interval of its next-hop and so on, until message reaches the sink node. Consequently, due to the autonomous same duration wake-up intervals, delays are added at each hop along the path to the sink, as each node has to wait for next-hop wake interval before transmitting packet. All these delays at each hop contribute to the final end-to-end delay of packet. This increase in delay is equal to the product of the number of intermediate forwarders times the length of the wakeup interval [17]. This delay is not acceptable for many delay-sensitive applications, which include military surveillance, tsunami alarm, smart hospitals, seismic detection, biomedical health monitoring, hazardous environment sensing, fire detection, intrusion detection, disaster monitoring, and real-time control, which require the event reporting delay to be small. In this article, the delay minimization problem is investigated and sleep/wake scheduling scheme to optimally choose the variable wakeup interval of the nodes based on node's dissimilar traffic loads is proposed. The proposed protocol provides distributed and low-complexity solution to this problem.

### 3. The proposed scheduling scheme

#### 3.1. Problem description

Sleep/wake scheduling has been used to save energy and extend the network lifetime. Energy efficiency has inherent tradeoff with delay, thus, generally in such sleep/wake scheduling strategies, maximization in network lifetime is achieved at the expense of increase in delay. In many delay-sensitive applications where real-time response is required, such delays could not be tolerated. In this article, delay minimization problem in delay-sensitive applications is addressed.

The first challenge for minimizing the expected end-to-end delay is to identify different areas, where extra delay can be minimized. To achieve this, delay minimization at three levels is analyzed and addressed: the delay occurred because of traffic load at the nodes near the BS, the delay occurred due to traffic load at the connectivity critical node, and delay occurred while dealing with traffic burst when an event occurs. It is identified that delay is incurred because of following three reasons. Firstly, during sleep/wake scheduling the wakeup requirement of nodes according to their position with respect to BS is

not considered. For instance, nodes near to the BS should have greater wake period as they are doing the data forwarding jobs on behalf of other nodes as well as send their own data. Secondly, nodes have the same awake interval throughout the network. However, the wakeup requirement in terms of forwarding traffic load is different for different nodes according to their topological importance in connectivity. For instance, a node which is cut vertex node and happens to connect two distant parts of network, is expected to handle more traffic load as compared with normal node. Thirdly, sleep scheduling is not adaptive to handle traffic burst on the occurrence of an event. When an event occurs, traditionally the wake interval of the node and its neighborhood remains same, whereas, in event occurrence, the node is expected to receive more traffic.

In this article, these challenges are addressed and delay is minimized at these three levels by using variable sleep/wake scheduling approach.

### 3.2. Proposed solution

To reduce the delay, a heuristic which maximizes the wakeup-up time in a scheduling period at three levels is used. The following is the brief description how this problem is addressed. Firstly, in a typical WSN architecture, all the nodes send their data to the sink node where the nodes near the sink nodes have to handle relatively more traffic. Sleep/wake scheduling disregards the fact that most packets go through the nodes near the sink node results in deteriorated performance. This article proposes that this delay can be minimized by considering the fact that forwarding requirement of the nodes is different according to their distance from the sink node. The sleep/wakeup schedule is directly related to the forwarding job, that is, more is the forwarding job more should be duration of wake interval. This is to minimize the schedule misses and to efficiently do the forwarding job with minimum delay. A sensor node that is near to the sink node is put into sleep state with lesser probability, and a sensor node that is away from the sink node is put into sleep state with greater probability. Consequently, the wake interval of the nodes increases as nodes come closer to the sink to handle the extra delay.

Secondly, because of the multi-hop communication paradigm of WSNs, a node's role in routing is important. Based on topology different nodes have different significance in the network. For instance, a scenario where there is only one node acting as a bridge between two distinct parts of the networks will have to forward all the traffic of one part of the network (depending upon location of the sink node). Thus, delay can be minimized by allocating sleep/wake schedule to the nodes according to the traffic load determined by the node's importance in connectivity. Giving a higher wake interval to heavily

loaded nodes (connectivity critical nodes) to ensure their availability when they are needed and giving a lower wake interval to lightly loaded nodes (less connectivity critical nodes) to save their energy.

Thirdly, when an event occurs at any particular area in a WSN, generic sleep/wake cycles of the nodes remain the same regardless of the frequency of the event detection. It does not adapt itself based on frequency and location of events in terms of changing their sleep wake interval. For this problem simple ideas of temporal and spatial dependency are used. Temporal dependency in this context refers that when an event occurs in sensing area of the node in one time slot, it is likely to occur in the proceeding time slots. Thus, if the nodes can adapt and change its sleep cycle, it can reduce the delay. Similarly, local dependency refers to the fact that, if an event occurrence is reported by sensor node, there is a likelihood of event occurrence in its neighborhood nodes. Thus, nodes in the neighborhood of that node should adapt to that traffic burst and change its sleep cycle. Thus, based on temporal dependency, the wake interval of node where event occurs is increased while based on spatial dependency; the wake interval of its neighbors is increased in the next time slot. These measures can significantly reduce the delay.

### 3.3. The proposed protocol description

In this article, sleep scheduling algorithm is proposed for event-driven sensor networks for delay-sensitive applications where events occurs rarely. The protocol consists of two main phases: the setup phase and the operation phase, as shown in Figure 2. These phases are further divided into sub-phases, as shown in Figure 3. The flow chart and the interaction among different phases are detailed in Figure 4. Different phases are discussed in the following paragraphs.

#### 3.3.1. Setup phase

Setup phase is divided into two sub-phases: initialization and route update.

- (1) *Initialization*: Each node computes its energy level and position in the networks. This information is used in sleep/wakeup scheduling, route update, and event reporting. Furthermore, the sink node divides the network into different regions. The sink node sends message to all the nodes in the network using three different transmission power (TP) where  $TP_1 < TP_2 < TP_3$ .  $TP_1$  defines region 1,  $TP_2$  defines region 2, and  $TP_3$  defines region 3 as shown in Figure 5. The sink node first propagates a beacon message with transmission power  $TP_1$ . The node receiving this message will mark its region status as region 1 and will go to sleep state. Next, the sink node again propagates a control message with transmission power  $TP_2$ . As nodes in region 1 are in sleep state, thus they

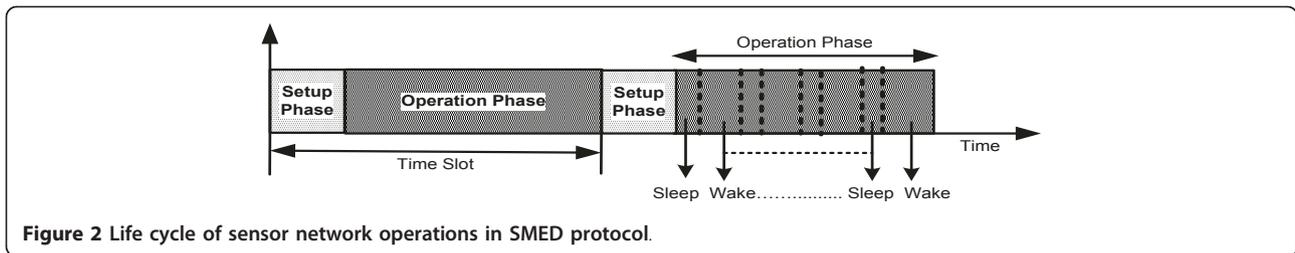


Figure 2 Life cycle of sensor network operations in SMED protocol.

will not receive this message. All the other nodes receiving this message will mark their location status as region 2 and will go into the sleep state. Similarly rest of the nodes mark their region status as region 3 by receiving message TP<sub>3</sub>.

The region information is retained by the nodes. Each node sends control message to maintain first hop neighbor information. Once a node has its neighbor information, it decides whether it is a connectivity critical node or not. To compute this, cut vertex method is used as adopted in Ref. [34]. Based on this computation, it will mark itself as a connectivity critical node; otherwise, it will mark itself as normal node.

(2) *Route update*: In the route update phase, the sink node generates a route discovery message with hop count 0 that is broadcasted throughout the network. A node upon receiving this broadcast message updates its hop count value, that is, changes its value to new value if received hop count value is less than previous hop count value, otherwise, retains the previous value. Before forwarding the route discovery message, each node increments the hop count and then broadcasts the message to nodes in its communication range. In this way, a message arrives at each node along the desired minimum cost path. Consequently, each node has a minimum hop count path to the sink node.

### 3.3.2. Operation phase

This phase is sub-divided into the sleep wake scheduling and the event reporting phase.

(1) *Sleep wake scheduling*: In this phase, sleep wake scheduling is performed based on traffic loads.

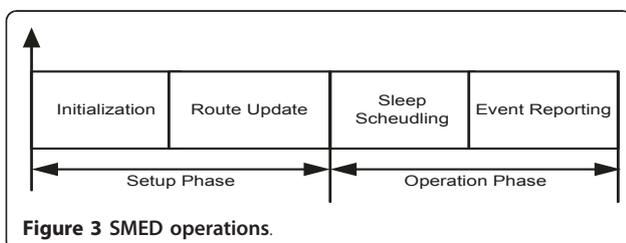


Figure 3 SMED operations.

Traffic load of nodes differ according to region they lay, their connectivity importance, and their proximity to the event occurrence. Based on these factors, each node determines its sleep wake pattern and accordingly switches between sleep/wake states (see Figure 6). In each sleep/wake cycle, node wakes up at set time intervals, waits for events to occur, scans the medium, and senses/receives data. The algorithm assigns three different wake interval (WT) WT<sub>1</sub>, WT<sub>2</sub>, and WT<sub>3</sub> to nodes in region 1, region 2, and region 3, respectively, where WT<sub>1</sub> < WT<sub>2</sub> < WT<sub>3</sub>. The wake interval of the node is inversely proportional to the distance from the sink node, that is, lesser the distance, the greater the wake interval (see Figure 7). It is done to make the wake interval adaptive to the traffic load of the nodes. Nodes near to the sink node have greater traffic load as compared with the nodes away from the sink node and are assigned longer wake intervals as shown in Figure 8. Also, connectivity critical nodes are assigned longer wake intervals to cater for the heavy traffic load. Algorithm 1 (see Appendix), explains how each node do the sleep/wakeup scheduling.

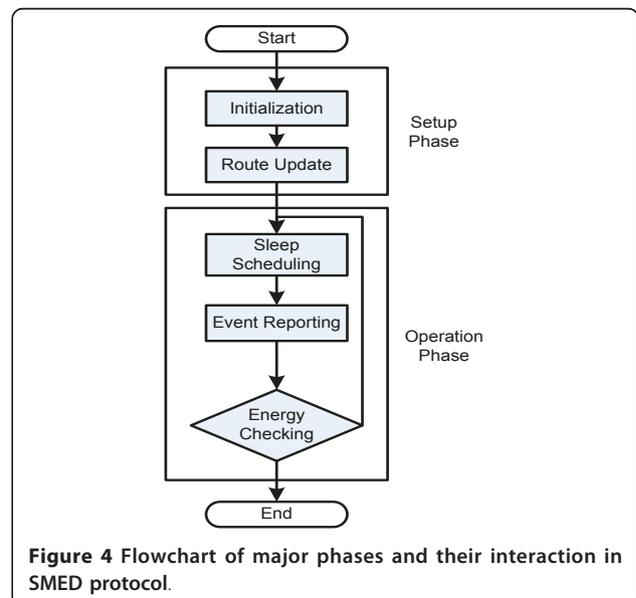
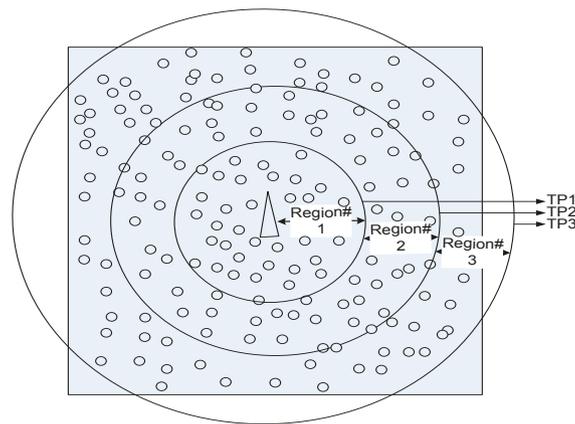


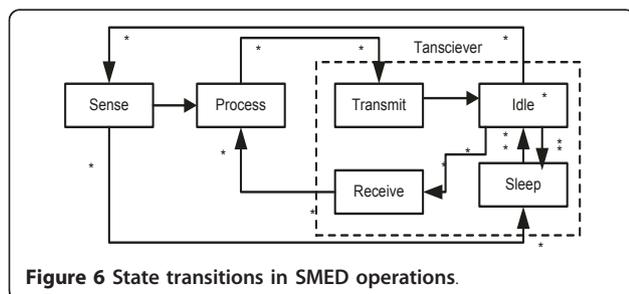
Figure 4 Flowchart of major phases and their interaction in SMED protocol.



**Figure 5** Dividing WSN into three regions.

(2) *Event reporting*: The event reporting phase is responsible for forwarding data to the sink node on the occurrence of event in timely manner. In this phase, data is gathered from the sensor nodes and sent to the sink node. When an event occurs in node's proximity, that node will increase its wake interval for the proceeding time slot. Furthermore, it sends message to its neighboring nodes to increase their wake interval to handle the expected traffic burst. This is because of the fact that when an event occurs in the node there is the probability of an event occurrence in the future as well (temporal dependency). Similarly if an event occurs at a node there is likelihood of occurrence of event in its neighborhood (spatial dependency). Both these case will result in expected traffic burst. Thus, the wake interval of a node where an event occurs and a node in its neighborhood is increased, to ensure the least delay in handling expected burst traffic (see Figure 9). Algorithm 2 (see Appendix), explains how each node do the sleep/wakeup scheduling based on the event occurrence.

To summarize, the flow chart of the SMED scheme is presented in Figure 10 for better understanding of the readers.



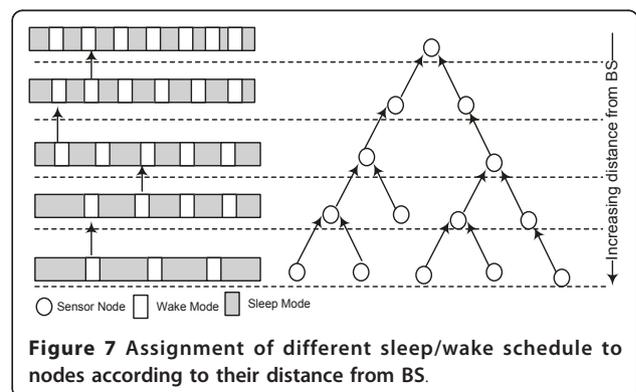
**Figure 6** State transitions in SMED operations.

#### 4. Simulations And Results

Based on the developed system model, simulations are carried out using OMNet<sup>++</sup> [22] to evaluate the performance of the proposed SMED protocol. Performance of proposed protocol is compared with the two contemporary protocols: S-MAC [17] and Anycast protocol [33]. The following are the details of the simulation setup, energy model, and discussion of the results.

##### 4.1 Simulation setup

Simulations were conducted in the sensing area of  $200 \times 200 \text{ m}^2$  and the number of sensor nodes varied from 20 to 260 for different experiments. Sensor nodes were randomly deployed and the random deployment is achieved by choosing  $(x, y)$  locations based on a uniform distribution. The sink node is fixed and located at the center of the network. The simulations were conducted with communication range equal to double of their sensing range. The simulation parameters are adopted as used in FlexiTP [8] which are based on Mica2Mote hardware. See the Table 1 for the listing of the simulation parameters.



**Figure 7** Assignment of different sleep/wake schedule to nodes according to their distance from BS.

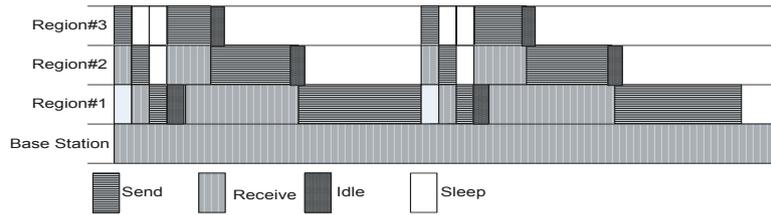


Figure 8 The path wakeup pattern of nodes in three regions.

#### 4.2 Event summary

The following is the event summary of the main events as used in SMED protocol (see Table 2).

#### 4.3. Energy model

It is assumed that the sensor nodes have the ability to adjust their transmission power according to the distance of the receiving node. The energy model

presented in [10] is adopted here. The amount of energy consumed for transmission ETX of an  $l$ -bit message over a distance is given by

$$E_{Tx} = \begin{cases} lE_{elect} + l\epsilon_{fs} \cdot d^2 & \text{for } 0 \leq d \leq d_{crossover} \\ lE_{elect} + l\epsilon_{mp} \cdot d^4 & \text{for } 0 \geq d_{crossover} \end{cases} \quad (1)$$

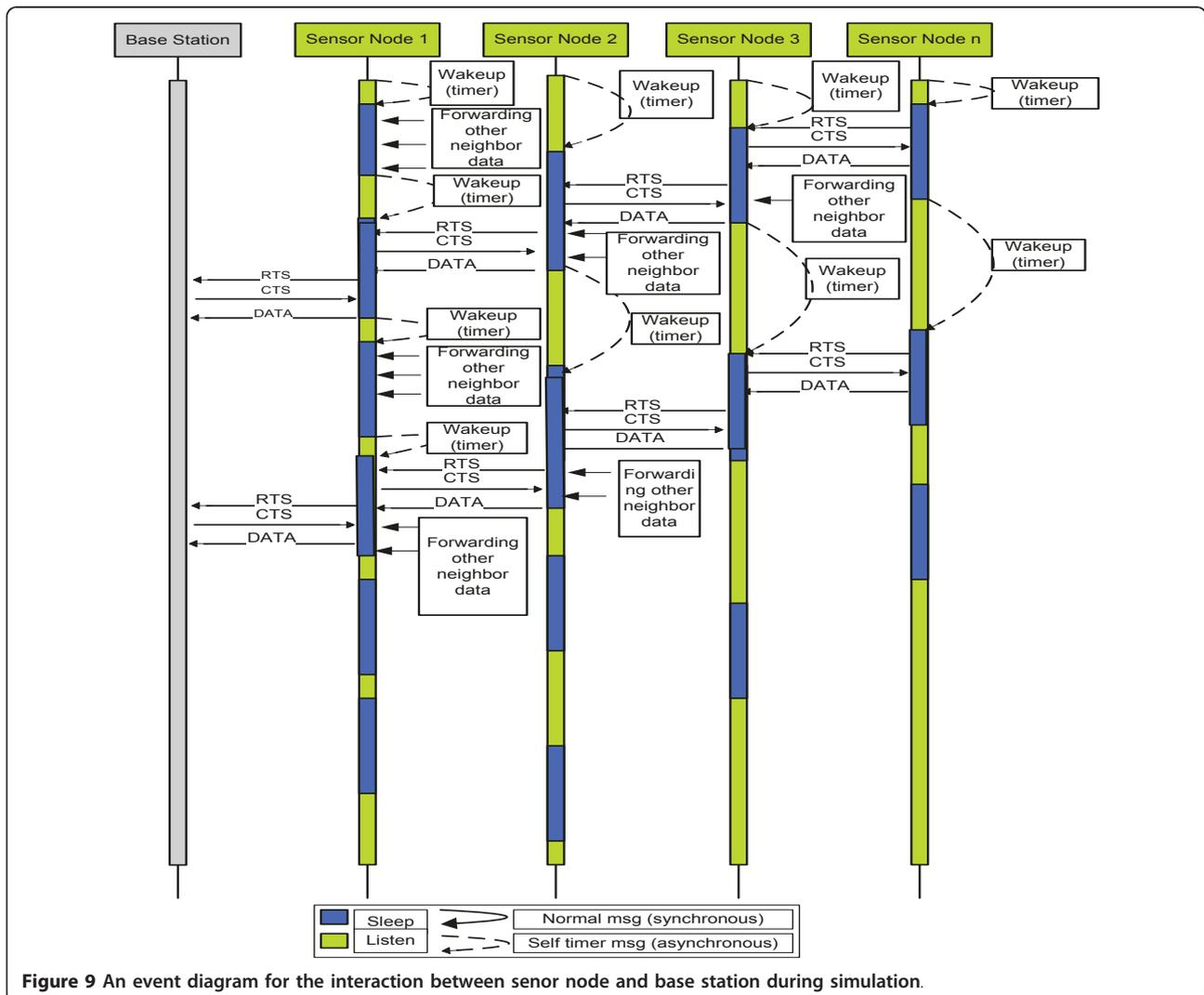


Figure 9 An event diagram for the interaction between sensor node and base station during simulation.

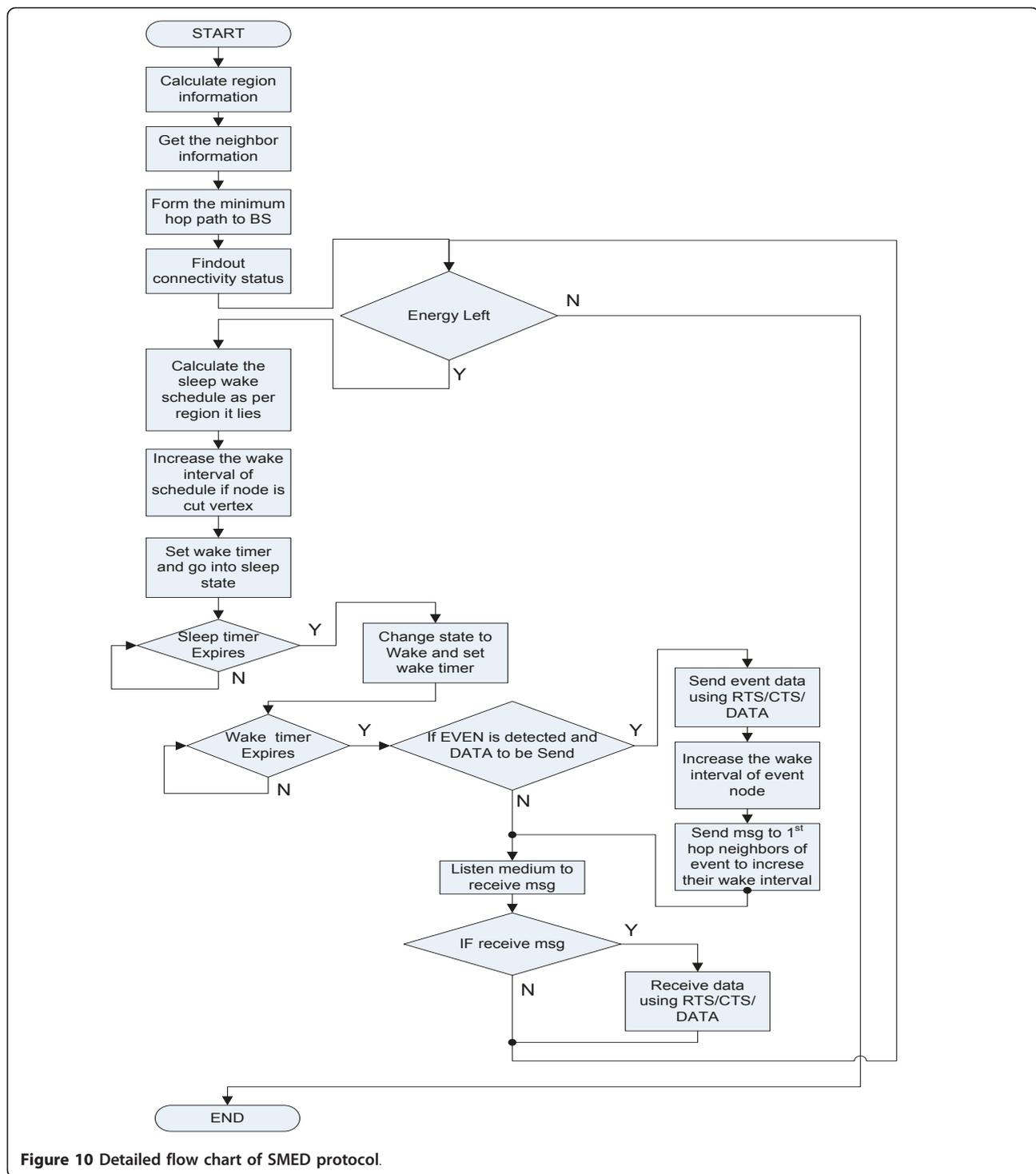


Figure 10 Detailed flow chart of SMED protocol.

where  $E_{elect}$  is the amount of energy consumed in electronics,  $\varepsilon_{fs}$  is constant for free space propagation and is the energy consumed in an amplifier when transmitting at distance shorter than  $d_{crossover}$ , and  $\varepsilon_{mp}$  is constant for multi-path propagation and is the amplifier energy consume in an amplifier when transmission at

distance greater the  $d_{crossover}$ .  $d$  is a distance threshold that depends on the environment. The energy expended in receiving  $l$ -bit message is given as

$$E_{Rx} = lE_{elect} \quad (2)$$

**Table 1 Simulation parameters for SMED protocol**

Simulation parameter	Value
Sensing area	200 × 200 m <sup>2</sup>
Bandwidth	38.4 Kbps
Transmission range	50 m
Transmit mode power	60 mW
Receive mode power	30 mW
Idle mode power	30 mW
Sleep mode power	0.003 mW
Transition power	30 mW
Transition time	2.45 ms
Packet size	96 bytes
Time slot size	42 ms
Simulation time	300 s

The total energy consumption is

$$E_{\text{radio}} = E_{\text{Tx}} + E_{\text{Rx}} \quad (3)$$

#### 4.4. Results and discussion

The performance of SMED is compared against the S-MAC and Anycast protocols. Experimental parameters, such as average delay per packet, energy per packet, average packet loss, and throughput, are used to measure the performance of SMED.

##### 4.4.1. Average delay per packet

Delay is referred to as the time span between the packet sent from a sensor node and packet received at the sink node. Delay values are measured by changing the number of sensor nodes from 20 to 260. As shown in Figure 11, the average delay experienced by the proposed SMED protocol is the least, while Anycast being the second and SMAC has the worst delay time. In SMED, nodes are given different wake intervals according to their traffic requirement with respect to their position in network, their topological importance and their proximity from the event. The proposed protocol is able to

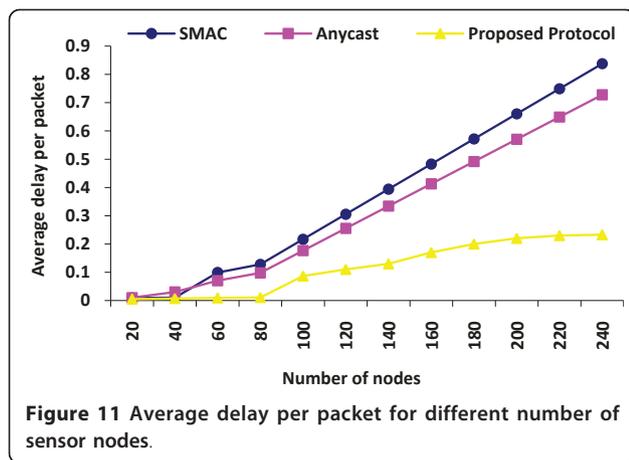
minimize delay at each hop because nodes have not to wait long for the wakeup interval of the next hop. As a result, average delay per packet in SMED is less than Anycast protocol and S-MAC. In the Anycast protocol, though the node has multiple next-hop relaying nodes by virtue of Anycast packet-forwarding scheme, which help to find next hop neighbor in quick manner, but still it does not consider the varied traffic requirement of different nodes. Thus, node wait time increases as packet approach to the nodes near to the sink node. Therefore, it has greater delay than the proposed protocol. In S-MAC protocol, nodes have fixed wake interval for the whole network irrespective of their traffic requirement, thus, each node has to wait for the wake interval of the next hop. However, as all nodes have to relay their data all the way to sink node using multi-hop communication. Hence, it involves many relay nodes to reach the sink node, which increases end-to-end delay. Considering the local traffic at each node, sensed data had to wait for sometime at each node to get attended, thus delay become longer when packet approaches the nodes near to sink. The problem got worse when the packet approaches the nodes near to the sink node, where packet suffers maximum delay. It makes the Anycast protocol and S-MAC prone to longer delays. Furthermore, as the number of nodes increases, the SMED clearly outperforms the other two strategies. For SMED, the performance remains the same for the increased number of nodes, since wake interval is adaptive to the traffic load, whatever may be the size of the network. In this way, increasing the node number has no effect on SMED. Therefore, it suggests that the proposed SMED protocol is more scalable than S-MAC and Anycast protocol.

##### 4.4.2. Average energy per packet

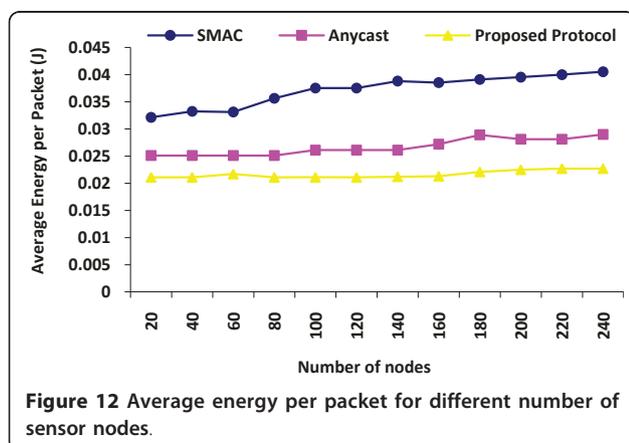
Average energy per packet is a measure of energy spent for forwarding a packet to the sink node. It is an indicator of the lifetime that can be achieved by the protocols.

**Table 2 Events summary for SMED**

S. no	event name	Description
1.	Initialization()	This event is executed in the start, when node is initialized. It involves initialization of node parameter, some of which includes nodeID, initial residual energy, random location coordinates etc.
2.	SN_W	This is activated when a node's self timer expires. Sensor node then changes its state from sleep to active state
3.	SN_S	This is activated when a node finishes sending/receiving data or self timer expires. Sensor node then changes its state from active to sleep state
4.	E_DETECT	This event is activated when a node sense any event. It is a self message
5.	SEND_MSG	This event is activated when a node receives self message event E_DETECT. It involves sending send the data to the next hop neighbor using RTS/CTS/DATA
6.	RECEIVE_MSG	This event is activated when a node receives any message RTS/CTS/DATA. It involves establishing the communication with sender node and receives the data using RTS/CTS/DATA
7.	End Timer	An event triggered when the WAIT timer expires. It can be at the completion of any of timer event detection, sleep or wake timer



In Figure 12, average energy per packet is plotted on  $y$ -axis, with varying number of sensor nodes (from 20 to 260) on  $x$ -axis. It can be observed that the average energy consumption per packet for the proposed SMED protocol is less than Anycast protocol and SMAC indicating comparatively extended network lifetime. The reason for increased lifetime in SMED can be attributed to the fact that it adjusts wake intervals based on traffic loads. By doing so, SMED avoids the case where the nodes remain awake and stay idle as no traffic is to be forwarded. Whereas, in the Anycast protocol many nodes stay awake to provide alternate paths for routing and mostly they remain idle, as expected traffic requirements of nodes are considered while setting up sleep/wake schedule. It results in increasing the wake node staying idle, which significantly limit the network lifetime. Similarly in SMAC protocol random sleep/wake schedule is defined for all the nodes which increase the number of wake idle nodes especially as move away from the sink node. Nodes away from the sink node have to do less relaying. Ultimately, it uses the energy of the nodes in idle listening and ultimately the network lifetime of the network is reduced. Hence proposed



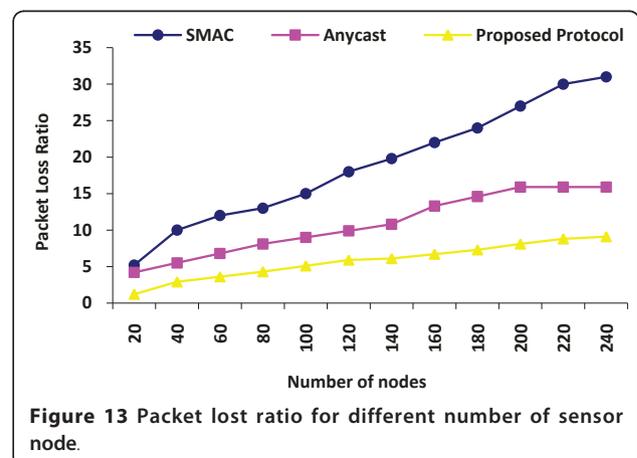
protocol has less energy per packet for both Anycast and SMAC protocol.

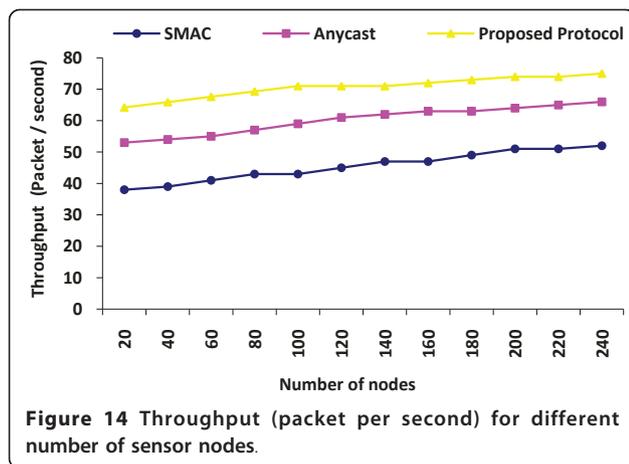
#### 4.4.3. Packet loss ratio (%)

Packet loss ratio refers to the percentage of the packets that could not reach the BS, that is,

$$\text{Packet loss ratio} = \frac{\text{Total number packets not received at the BS}}{\text{Total number packets send by all the sensor nodes}} \quad (4)$$

Figure 13 shows the measurement of packet loss ratio for the three protocols with varying number of sensor nodes (20 to 240). The packet loss ratio of all protocols increases as the number of nodes increases but their slopes are different. The packet loss ratio of the proposed protocol increases at the slowest rate because it considers the traffic pattern of different nodes and accordingly assigns wake intervals, which result in less packet loss. For S-MAC the packet loss ratio grows at relatively high speed because it use random sleep/wake schedule, which increases schedule misses and ultimately increases the packet loss ratio. It is clear from the Figure 13 that for the whole simulation, SMED has a far lower packet loss ratio as compared with S-MAC and Anycast protocols. It is because in the S-MAC and Anycast protocols, there is no congestion control mechanism to ensure adaptability in the wake interval, thus, the number of collisions/misses is higher. It makes S-MAC and Anycast more vulnerable to packet loss, thus reducing transmission reliability. With SMED, nodes have adaptive wake interval according to their traffic load requirement, thus, traffic flows through the network smoothly. Furthermore, in SMED critical nodes in terms of connectivity have longer wake interval, which results in decreased packet loss. Node detecting event and node in its vicinity are assigned greater wake interval, which also contribute in minimizing the lost packets. Hence, SMED is less prone to packet loss than other two protocols. As a result, SMED protocol has





outperformed the other two protocols in terms of packet lost ratio.

#### 4.4.4. Throughput (packets per second)

Throughput is measured by the number of packets received per second at the sink node. In this simulation, the number of nodes were varied from 20 to 260 and the throughput is measured at the sink node. It can be seen from Figure 14 that by increasing the number of sensor nodes, the throughput for SMED, S-MAC, and Anycast protocols have increased. For greater network size after that, SMED has achieved throughput greater than both the S-MAC and Anycast protocols. It showed that the SMED protocol is scalable and could perform better as the size of the WSN becomes larger. Simply, out of the three evaluated protocols, SMED has the best throughput, while S-MAC second, and Anycast protocol has the least.

#### 4.4.5. Coverage lifetime

Coverage lifetime is referred as the time the network is able to preserve 100% or over 90% coverage of the whole

sensing area. As a generalization, coverage of less than this percentage is not tolerable and can be regarded as failure. In this experiment, number of nodes is varied from 20 to 260, and coverage lifetime (100%, 90%) is measured. Figure 15a provides coverage lifetime for 100% coverage area, while Figure 15b depicts coverage lifetime for 90% coverage area. It can be observed that proposed outperformed Anycast and SMAC in terms of coverage lifetime in both cases of 100% and 90% coverage. In the SMAC approach, nodes used randomly do the sleep/wake scheduling. It increases the idle listening and involves more schedule miss because while defining sleep/wake schedule traffic requirement of the nodes are not considered. In Anycast protocol more than one paths are made active to have smooth routing. It wastes the energy of some of the nodes in idle listening and schedule misses because in making nodes active, traffic requirement of the nodes is not considered. These nodes soon expire their energy and appear as coverage hole in the network. Whereas, in SMED, coverage hole is avoided by having efficient use of nodes energy by assigning nodes sleep/wake schedule according to its traffic requirement with respect to their position in network, their topological importance and their proximity from the sink node.

## 5. Conclusion

In this article, we have proposed the SMED in Multi-hop Wireless Sensor Networks. The proposed scheme is designed to have variable active duration of nodes according to their variable traffic load. The variable active durations are assigned to the nodes based on node distance from the sink node, node topological importance, and occurrence of event in its vicinity. It will enable the nodes to gracefully handle the traffic, as nodes are dynamically assigned active durations according to their

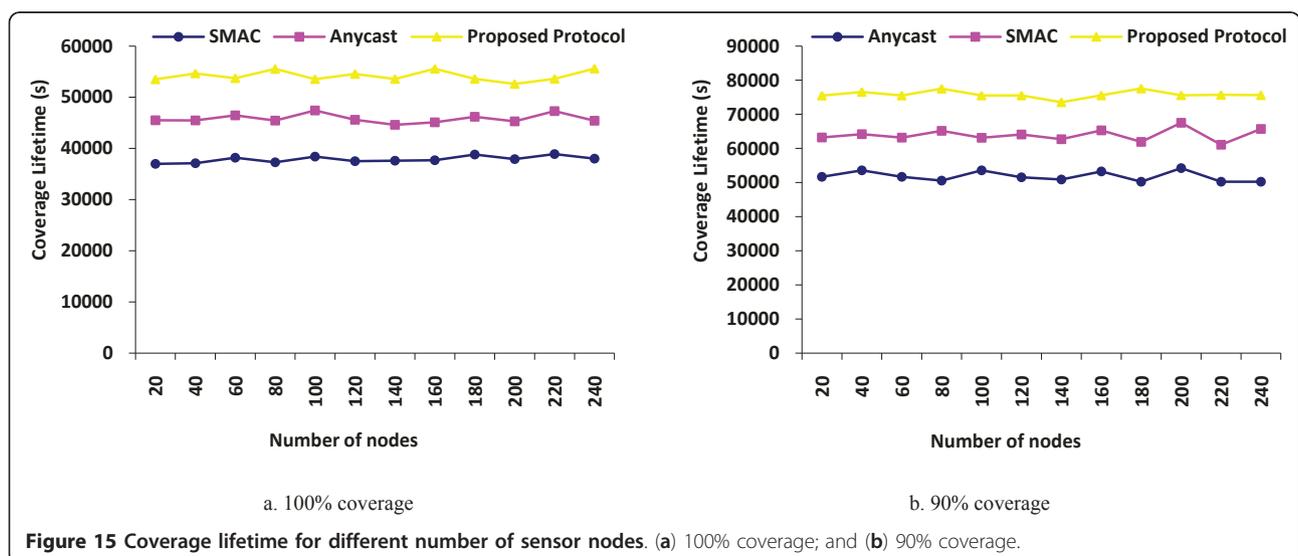


Figure 15 Coverage lifetime for different number of sensor nodes. (a) 100% coverage; and (b) 90% coverage.

expected traffic load. It minimizes delay at the nodes near to the sink node, node having critical topological position, and nodes in vicinity of event occurrence. This ensures rapid dissemination of data to the sink node and hence reduces the end-to-end delay. Simulations are carried out to evaluate the performance of the proposed protocol, by comparing its performance with S-MAC and Anycast protocols. Simulation results demonstrated that the proposed protocol has significantly reduced the end-to-end delay, as well as improved the other QoS parameters of average energy per packet, average delay, packet loss ratio, throughput, and coverage throughput.

In future our plan is to extend the simulations to consider other parameters and scenarios, such as fault tolerance, impact of aggregation, etc. Other important future extension is to evaluate the performance of SMED in clustered-based WSNs.

## Appendix

### Algorithms 1 and 2

**Algorithm 1:** sleep/wakeup scheduling algorithm

**Input:**  $WI_{normal}$  the Wake Interval Normal

**Output:** sleep/wakeup schedule

**begin**

**check** region status by each node

**if** *region status is region 1* **then**

    set wake interval to  $3 \times WI_{normal}$

**else**

**if** *region is region 2* **then**

        set wake interval to  $2 \times WI_{normal}$

**else**

    set wake interval to  $WI_{normal}$

**endif**

**endif**

**check** topology status by each node

**if** *topology status is critical* **then**

    set wake interval to  $2 \times WI_{normal}$

**else**

    set wake interval to  $WI_{normal}$

**endif**

**follow up**, by each sensor node, the determined sleep/wake schedule and sense the environment for the occurrence of the event for the specified interval defined by sleep/wake schedule.

**if** event occurs **then**

    set wake interval of event occurrence node to  $3 \times WI_{normal}$

    set wake interval first hop neighbors of event occurrence node to

$2 \times WI_{normal}$  and send the message about the changed wake interval to the first hop neighbors

**change** of sleep/wake interval by the first hop neighbor upon receipt of the updated sleep/wake schedule

**wait** of the event occurrence node for the arrival of next scheduled slot determined by self timer before sending the sensed event data

**send** the data to the next hop neighbor using three way communication RTS, CTS and DATA.

**endif**

**end**

**Algorithm 2:** sleep/wakeup scheduling algorithm

**Input:**  $WI_{normal}$  the Wake Interval Normal

**Output:** sleep/wakeup schedule

**begin**

**follow** the determined sleep/wake schedule and sense the environment for the occurrence of the event for the specified interval defined by sleep/wake schedule and employed by self timer

**if** event occurs **then**

**set** wake interval of event occurrence node to  $3 \times WI_{normal}$

**set** wake interval of the first hop neighbors of event occurrence node to  $2 \times WI_{normal}$  and send the message about the changed wake interval to the first hop neighbors

**change** of sleep/wake interval by the first hop neighbor upon receipt of the updated sleep/wake schedule

**wait** of the event occurrence node for the arrival of next scheduled slot determined by self timer before sending the sensed event data

**send** the data to the next hop neighbor using three way communication RTS, CTS and DATA.

**go** into sleep mode and wakeup in next schedule interval: a node will wakeup in next schedule slot determined by updated sleep/wake schedule.

**endif**

**end**

### Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. The authors

would also like to thank Universiti Teknologi PETRONAS for providing the financial assistance in the form of the Graduate Assistantship (GA) to carry out the research.

#### Author details

<sup>1</sup>Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia <sup>2</sup>COMSATS Institute of Information Technology (CIIT), Tobe Camp, University Road, 22060 Abbottabad, Pakistan

#### Competing interests

The authors declare that they have no competing interests.

Received: 5 March 2011 Accepted: 9 September 2011

Published: 9 September 2011

#### References

1. S Gyula, Maróti Miklós, L'étédeczi Ákos, B György, K Branislav, Nadas András, P Gábor, S János, F Ken, Sensor network-based countersniper system, in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, ACM, Baltimore, (2004)
2. M Castillo-Effer, DH Quintela, W Moreno, R Jordan, W Westhoff, Wireless sensor networks for flash-flood alerting, in *Devices, Circuits and Systems* (2004). in *Proceedings of the Conference on Fifth IEEE International Caracas K Lorincz, DJ Malan, TRF Fulford-Jones, A Nawoj, A Clavel, V Shnayder, G Mainland, M Welsh, S Moulton, Sensor networks for emergency response: challenges and opportunities*, IEEE Pervas. Comput. **3**(4), 16–23 (2004)
3. IF Akyildiz, S Weilian, Y Sankarasubramaniam, E Cayirci, A survey on sensor networks. *IEEE Comm Mag.* **40**(8), 102–114 (2002). doi:10.1109/MCOM.2002.1024422
4. Y Jennifer, M Biswanath, G Dipak, Wireless sensor network survey. *Comput Netw.* **52**(12), 2292–2330 (2008). doi:10.1016/j.comnet.2008.04.002
5. W Andrew, C SeongHwan, S Charles, C Anantha, Energy efficient Modulation and MAC for Asymmetric RF Microsensor Systems, in *Proceedings of the 2001 international symposium on Low Power Electronics and Design*, ACM, Huntington Beach, CA, (2008)
6. BA Myers, JB Willingham, P Landy, MA Webster, P Frogge, M Fischer, Design considerations for minimal-power wireless spread spectrum circuits and systems. *Proc IEEE.* **88**(10), 1598–1612 (2000). doi:10.1109/5.888998
7. WL Lee, A Datta, R Cardell-Oliver, Flexi TP, a flexible-schedule-based TDMA protocol for fault-tolerant and energy-efficient wireless sensor networks. *IEEE Trans Parallel Distrib Syst.* **19**(6), 851–864 (2008)
8. C Srisathapornphat, S Chien-Chung, Coordinated power conservation for ad hoc networks, in *IEEE International Conference on Communications* (2002)
9. NA Pantazis, DD Vergados, A survey on power control issues in wireless sensor networks. *IEEE Commun Surv Tutor.* **9**(4), 86–107 (2007)
10. AP Nikolaos, JV Dimitrios, DV Dimitrios, D Christos, Energy efficiency in wireless sensor networks using sleep mode TDMA scheduling. *Ad Hoc Netw.* **7**(2), 322–343 (2009). doi:10.1016/j.adhoc.2008.03.006
11. J Liu, S Singh, ATCP: TCP for mobile ad hoc networks. *IEEE J Sel Areas Commun.* **19**(7), 1300–1315 (2001). doi:10.1109/49.932698
12. L Jie, Z Feng, P Cheung, L Guibas, Apply geometric duality to energy-efficient non-local phenomenon awareness using sensor networks. *IEEE Wirel Commun.* **11**(6), 62–68 (2004). doi:10.1109/MWC.2004.1368898
13. C Juan-Carlos, M Pietro, A performance comparison of energy consumption for mobile ad hoc network routing protocols, in *Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, IEEE Computer Society, (2000)
14. V Tsaoussidis, H Badr, TCP-probing: towards an error control schema with energy and throughput performance gains, in *Proceedings of 2000 International Conference on Network Protocols* (2000)
15. S Suresh, CS Raghavendra, PAMAS—power aware multi-access protocol with Signalling for Ad Hoc Networks. *SIGCOMM Comput Commun Rev.* **28**(3), 5–26 (1998). doi:10.1145/293927.293928
16. Y Wei, J Heidemann, D Estrin, An energy-efficient MAC protocol for wireless sensor networks, in *Proceedings on Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM* (2002)
17. C Benjje, J Kyle, B Hari, M Robert, Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wirel Netw.* **8**(5), 481–494 (2002). doi:10.1023/A:1016542229220
18. L van Hoesel, T Nieberg, W Jian, PJM Havinga, Prolonging the lifetime of wireless sensor networks by cross-layer interaction. *IEEE Wirel Commun.* **11**(6), 78–86 (2004). doi:10.1109/MWC.2004.1368900
19. A Giuseppe, C Marco, F Mario Di, P Andrea, Energy conservation in wireless sensor networks: A survey. *Ad Hoc Netw.* **7**(3), 537–568 (2009). doi:10.1016/j.adhoc.2008.06.003
20. C Schurgers, V Tsiatsis, MB Srivastava, STEM: topology management for energy efficient sensor networks, in *Proceedings on Aerospace Conference* (2002)
21. C Schurgers, V Tsiatsis, S Ganeriwala, M Srivastava, Optimizing sensor networks in the energy-latency-density design space. *IEEE Trans Mobile Comput.* **1**(1), 70–80 (2002). doi:10.1109/TMC.2002.1011060
22. Y Xue, NH Vaidya, A wakeup scheme for sensor networks: achieving balance between energy saving and end-to-end delay, in *10th IEEE Proceedings on Real-Time and Embedded Technology and Applications Symposium, RTAS 2004* (2004)
23. T Yu-Chee, H Chih-Shun, H Ten-Yueng, Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks, in *Proceedings on Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2002* (2002)
24. Z Rong, CH Jennifer, S Lui, Asynchronous wakeup for ad hoc networks, in *Proceedings of the 4th ACM International Symposium on Mobile ad hoc Networking & Computing*, ACM, Annapolis, MD, (2002)
25. D Jing, SH Yunghsiang, BH Wendi, KV Pramod, Scheduling sleeping nodes in high density cluster-based sensor networks. *Mobile Netw Appl.* **10**(6), 825–835 (2005). doi:10.1007/s11036-005-4441-9
26. D Jing, SH Yunghsiang, BH Wendi, KV Pramod, Balanced-energy sleep scheduling scheme for high-density cluster-based sensor networks. *Comput Commun.* **28**(14), 1631–1642 (2005). doi:10.1016/j.comcom.2005.02.019
27. Y Xu, J Heidemann, D Estrin, Adaptive energy-conserving routing for multihop ad hoc network. (2000) Research Report 527, USC/Information Sciences Institute
28. T Di, DG Nicolas, A coverage-preserving node scheduling scheme for large wireless sensor networks, in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, ACM, Atlanta, GA, (2002)
29. P Lin, C Qiao, X Wang, Medium access control with a dynamic duty cycle for sensor networks. in *Wireless Communications and Networking Conference* (2004)
30. D Tijs van, L Koen, An adaptive energy-efficient MAC protocol for wireless sensor networks, in *Proceedings of the 1st international conference on Embedded networked sensor systems*, ACM, Los Angeles, CA, (2003)
31. Y Shih-Hsien, T Hung-Wei, EH Wu, C Gen-Huey, Utilization based duty cycle tuning MAC protocol for wireless sensor networks, in *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE* (2005)
32. K Joohwan, L Xiaojun, NB Shroff, P Sinha, Minimizing delay and maximizing lifetime for wireless sensor networks with anycast. *IEEE/ACM Trans Netw.* **18**(2), 515–528 (2010)
33. AA Abbasi, K Akkaya, M Younis, A distributed connectivity restoration algorithm in wireless sensor and actor networks, in *32nd IEEE Conference on Local Computer Networks 2007, LCN 2007* (2007)

doi:10.1186/1687-1499-2011-92

Cite this article as: Nazir et al.: Sleep/wake scheduling scheme for minimizing end-to-end delay in multi-hop wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:92.